GOOGLE COLAB:
CODE:

```
# Mount Google Drive to access the dataset
from google.colab import drive
drive.mount('/content/drive')
# Install necessary libraries
!pip install tensorflow
# Import libraries
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.models import Sequential
# Define paths to the dataset
base_dir = '/content/drive/MyDrive/Colab
Notebooks/forest_fire_detection/dataset'
train_dir = os.path.join(base_dir, 'train')
validation_dir = os.path.join(base_dir, 'validation')
# Preprocess and augment data
train_datagen = ImageDataGenerator(
rescale=1./255,
rotation_range=40,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True,
fill_mode='nearest'
)
validation_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(
train_dir,
target_size=(150, 150),
batch_size=20,
class_mode='binary'
)
validation_generator = validation_datagen.flow_from_directory(
validation_dir,
target_size=(150, 150),
batch_size=20,
class_mode='binary'
)
# Define the CNN model
model = Sequential([
Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
MaxPooling2D(2, 2),
Conv2D(64, (3, 3), activation='relu'),
MaxPooling2D(2, 2),
Conv2D(128, (3, 3), activation='relu'),
MaxPooling2D(2, 2),
Conv2D(128, (3, 3), activation='relu'),
MaxPooling2D(2, 2),
Flatten(),
Dense(512, activation='relu'),
Dense(1, activation='sigmoid')
])
# Compile the model
model.compile(loss='binary_crossentropy',
optimizer=tf.keras.optimizers.RMSprop(lr=1e-4),
metrics=['accuracy'])
# Train the model
history = model.fit(
train_generator,
steps_per_epoch=100,
epochs=20,
validation_data=validation_generator,
validation_steps=50
)
# Save the model
model.save('/content/drive/MyDrive/Colab
Notebooks/forest_fire_detection/forest_fire_detection_model.h5')
```

OUTPUT

```
Found 2000 images belonging to 2 classes. Found 1000 images belonging
to 2 classes. Epoch 1/20 100/100 [==============================] -
35s - loss: 0.6942 - accuracy: 0.5070 - val_loss: 0.6910 -
val_accuracy: 0.5010 Epoch 2/20 100/100
[==============================] - 34s - loss: 0.6918 - accuracy:
0.5275 - val_loss: 0.6895 - val_accuracy: 0.5340 ... Epoch 20/20
100/100 [==============================] - 36s - loss: 0.4012 -
accuracy: 0.8195 - val_loss: 0.5620 - val_accuracy: 0.735
```